CS112 - Fall 2022

Lab10

Instructor: Paul Haskell

INTRODUCTION

In this lab, you will develop some Java programs that use different types of Java loops. It sure is nice to have the computer do the work to complete hundreds or thousands of calculations, without our having to do them by hand.

Nested for() loops

Create a program called **TextTriangle.java**. This program should use nested for() loops to print out the following pattern:

```
abcdefghijklmnopqrstuvwxyz
cdefghijklmnopqrstuvwxyz
efghijklmnopqrstuvwxyz
ghijklmnopqrstuvwxyz
…
```

Note that the first **two** characters are dropped when moving from one line to the next.

Square Root Finder

Your next program, **Root.java,** will calculate the square root of a positive number. But rather than using the `Math.sqrt()` method from the Math library, you will calculate it yourself using a Java loop.

Read the input from the first command-line argument. (Print "ERROR" to `System.err` if there is no argument or if the argument is negative.) Now implement the following square root algorithm:

- set two limits for the square root, called `low` and `high`. Initialize low to 0.0 and high to your input number
- calculate `avg` to be the average of `low` and `high`
- while the difference between `low` and `high` is greater than 0.1% of low, if $\texttt{avg}^2$ is greater than the original input number, then replace `high` with `avg`. If $\texttt{avg}^2$ is less than the original input number, then replace `low` with the average. Now update the average and continue.
- Once you have achieved the 0.1% threshold, print out `avg`

Easy! But your testing should show that this algorithm works fine for some numbers but not others. Can you find which numbers have difficulties? **OPTIONAL EXERCISE:** Can you extend this algorithm to handle the numbers that the simple original algorithm handles incorrectly?

An important result from finance

Have you heard the story of the Dutch buying Manhattan Island from the American Indians for $24? It is a true story (link). Four hundred years ago, Dutch explorers bought Manhattan Island from the American Indian locals for $24 worth of furs and beads. For a long time, people told this story with the sub-message, “Weren’t the Indians foolish?”

Recently it was discovered that the American Indians from whom the Dutch bought Manhattan actually did not live on Manhattan—they were enemies of the tribe that lived on Manhattan. The sellers were thinking, “These stupid Europeans can’t tell one tribe from another. If they want to give us stuff and kick our enemies out of Manhattan, that sounds good to us.”

But that’s not why I care about this story.

If the American Indians invested that $24 in a checking account earning 0.3% interest, for 396 years, today it would be worth $79. If they instead invested that money in a savings account earning 2% interest, today it would be worth $61,000. If that $24 were invested in the stock market, earning 8% per year on average, today it would be worth **a lot of money** (more than the current worth of all the real estate in Manhattan). 8% is an accurate figure for the US stock market annual returns (link).

Your assignment of course is to figure out how much money a $24 investment would be worth if invested for 396 years at 8% interest. You can approach the problem as follows:

- After 1 year, the investment value is worth (1.08) * $24
- After 2 years, the investment value is worth $(1.08)^2$ * $24
- etc

Write a program called **Manhattan.java** that calculates and prints out the value of the investment. Use a while() loop or for() loop to do the calculation, not any built-in Math library function. Your answer should only print the final value, not every line of calculation. Please start your answer with a '$' character, e.g.

```
$<<some big number>>
```

Reminder

Put all your files in Lab10 and push to GitHub before the deadline. This assignment must be turned in before Monday Oct 3rd at 11:59pm.

Conclusion

As you developed these programs you worked with several different types of Java loops, and of course got more practice developing and testing robust programs. Hopefully the financial results was unexpected or at least interesting.

Grading Rubric

TestTriangle.java and **Manhattan.java** are worth 5 points each, for correct output.

Root.java is worth 20 points: 2 points for each of 5 test cases.

- 0-10 points for the graders' evaluation of your software clarity and design quality